The Maximum Latency and Identification of Positive Boolean Functions

MAKINO, Kazuhisa; IBARAKI, Toshihide

数理解析研究所講究録
第 871 巻 1994 年 73-79

The Maximum Latency and Identification of Positive Boolean Functions
正論理関数の最大潜伏度と同定

Kazuhisa MAKINO and Toshihide IBARAKI
牧野和久　　茨木俊秀

Department of Applied Mathematics and Physics,
Faculty of Engineering,
Kyoto University

abstract

Consider the problem of identifying $\min T(f)$ and $\max F(f)$ of a positive (i.e., monotone) Boolean function f, by using membership queries only, where $\min T(f)$ ($\max F(f)$) denotes the set of minimal true vectors (maximal false vectors) of f. It is known that an incrementally polynomial algorithm exists if and only if there is a polynomial time algorithm to check the existence of an unknown vector for given sets $MT \subseteq \min T(f)$ and $MF \subseteq \max F(f)$. Unfortunately, however, the complexity of this problem is still unknown. To answer this question partially, we introduce in this paper a measure for the difficulty of finding an unknown vector, which is called the maximum latency. If the maximum latency is constant, then an unknown vector can be found in polynomial time and there is an incrementally polynomial algorithm for identification. Several subclasses of positive functions are shown to have constant maximum latency, e.g., 2-monotonic positive functions, Δ-partial positive threshold functions and matroid functions, while the class of general positive functions has maximum latency not smaller than $\lfloor n/4 \rfloor + 1$ and the class of positive k-DNF functions has $\Omega(\sqrt{n})$ maximum latency.

1 Introduction

Consider the problem of identifying $T(f)$ (set of true vectors) and $F(f)$ (set of false vectors) of a given Boolean function (or a function in short) f by asking membership queries to an oracle whether $f(u) = 0$ or 1 holds for some selected vectors u [2]. In the terminology of computational learning theory [1, 12], this is the exact learning of a Boolean theory f by membership queries only. It is also a process of forming a theory that explains a certain phenomenon by collecting positive and negative data (in the sense of causing and not causing that phenomenon) [4]. In particular, we are interested in the case where f is known to be positive, i.e., monotone. If f is a positive function, $T(f)$ and $F(f)$ can be compactly represented by $\min T(f)$ (set of minimal true vectors) and $\max F(f)$ (set of maximal false vectors). Therefore our problem is stated as follows.

Problem IDENTIFICATION

Input: an oracle for a positive function f.
Output: $\min T(f)$ and $\max F(f)$.

The complexity of this type of enumeration algorithm is usually measured in its length of input and output. An algorithm to enumerate items $a_1, a_2, \ldots, a_p$ is called *incrementally polynomial* [7], (i) if it iterates the following procedure for $i = 1, 2, \ldots, p$: output the i-th item a_i from the knowledge of its input and items $a_1, a_2, \ldots, a_{i-1}$ generated by then, and (ii) if the time required for the i-th iteration is polynomial in the input length and the sizes of $a_1, a_2, \ldots, a_{i-1}$. If an algorithm is incrementally polynomial, it also satisfies the criterion of *polynomial total time* [7] (i.e., polynomial time in the length of input and output).

Now let MT and MF respectively denote the partial knowledge of $\min T(f)$ and $\max F(f)$ currently at hand, i.e.,

$$MT \subseteq \min T(f) \text{ and } MF \subseteq \max F(f). \quad (1.1)$$

Define

$$T(MT) = \{v \mid v \geq w \text{ for some } w \in MT\}$$
$$F(MF) = \{v \mid v \leq w \text{ for some } w \in MF\}.$$

By assumption (1.1), $T(MT) \subseteq T(f)$ and $F(MF) \subseteq F(f)$, and hence

$$T(MT) \cap F(MF) = \emptyset$$

holds. A vector u is called *unknown* if

$$u \in \{0,1\}^n \setminus (T(MT) \cup F(MF)),$$

since it is not known at the current stage whether u is a true vector or a false vector of f. If there is no unknown vector, then $T(MT) \cup F(MF) = \{0,1\}^n$ holds, i.e., $MT = \min T(f)$ and $MF = \max F(f)$ hold for some positive function f.

The general procedure of identifying a positive function f can be described as follows.

Algorithm IDENTIFY
Input: an oracle for a positive function f.
Output: min$T(f)$ and max$F(f)$.

1. Start with appropriate sets MT ($\subseteq$ min $T(f)$) and MF ($\subseteq \max F(f)$), where $MT \cup MF \neq \phi$ is assumed.

2. Test if $T(MT) \cup F(MF) = \{0,1\}^n$ holds. If so, output MT and MF, and halt. Otherwise, find an unknown vector u and go to 3.

3. Ask an oracle if $f(u) = 1$ or $f(u) = 0$. If $f(u) = 1$, then compute a new minimal true vector y such that $y \leq u$ and let $MT := MT \cup \{y\}$. On the other hand, if $f(u) = 0$, compute a new maximal false vector y such that $y \geq u$ and let $MF := MF \cup \{y\}$. Return to 2. □

The crucial part of this algorithm is in Step 2, i.e., to solve the following problem, where a set of vectors M is *incomparable* if any pair of vectors v and w in M satisfies $v \not\geq w$ and $w \not\geq v$.

Problem EQ
Input: Incomparable sets MT, MF ($\subseteq \{0,1\}^n$) such that $T(MT) \cap F(MF) = \emptyset$.
Question: Does $T(MT) \cup F(MF) = \{0,1\}^n$ (i.e., no unknown vector) hold?

If problem EQ can be solved in polynomial time, it is known that an unknown vector in Step 2 can be found in polynomial time [2], and that computing a minimal true vector or a maximal false vector y from an unknown vector u in Step 3 can also be done in polynomial time [1, 2, 12]. Therefore, an incrementally polynomial algorithm exists if and only if problem EQ can be solved in polynomial time. It is shown in [2] that problem EQ is polynomially equivalent to many other interesting problems encountered in various fields such as hypergraph theory [5], theory of coteries (used in distributed systems) [6], artificial intelligence [11] and Boolean theory [2]. Unfortunately, the complexity of these problems is still open, though there is some evidence to conjecture that it is co NP-complete.

In order to investigate the complexity of EQ for subclasses of positive functions, we introduce in this paper the concept of maximum latency, which is a complexity measure for finding an unknown vector. If the maximum latency is constant, then it is shown in section 2 that EQ can be solved in polynomial time, though the converse is not generally true. In section 4, we show that the maximum latency of general positive functions is at least $\lfloor n/4 \rfloor + 1$. However, several special classes of positive functions are found in section 3 to have constant maximum latency; classes of (i) 2-monotonic positive functions [3, 10], (ii) Δ-partial positive threshold functions [10], (iii) matroid functions [13], (iv) k-tight functions, and (v) others. For these classes of positive functions, therefore, there are incrementally polynomial identification algorithms. Finally it is shown in section 4 that the class of positive k-DNF functions has the maximum latency of $\Omega(\sqrt{n})$, even though it is known [5] that EQ can be solved in polynomial time for this class of functions.

The last result indicates that the concept of maximum latency is not always sufficient to distinguish polynomially solvable cases from those not solvable in polynomial time. However, it is also evident that the maximum latency is a powerful tool to find polynomially solvable special cases.

2 Definitions and basic properties

A *Boolean function*, or a *function* in short, is a mapping $f : \{0,1\}^n \mapsto \{0,1\}$, where $v \in \{0,1\}^n$ is called a *Boolean vector* (a *vector* in short). If $f(v) = 1$ (resp. 0), then v is called a *true* (resp. *false*) vector of f. The set of all true vectors (false vectors) is denoted by $T(f)$ $(F(f))$. A function f is *positive* if $v \le w$ always implies $f(v) \le f(w)$. A positive function is also called *monotone*. A true vector v of f is *minimal* if there is no other true vector w such that $w < v$, and let $\min T(f)$ denote the set of all minimal true vectors of f. A *maximal* false vector is symmetrically defined and $\max F(f)$ denotes the set of all maximal false vectors of f.

If f is positive, it is known that f has the unique disjunctive normal form (DNF) consisting of all prime implicants. There is one-to-one correspondence between prime implicants and minimal true vectors. For example, a positive function $f = x_1x_2 \vee x_2x_3 \vee x_3x_1$, has prime implicants x_1x_2, x_2x_3, x_3x_1 which correspond to minimal true vectors (110), (011), (101), respectively. The input length to describe a positive function f is $O(n|\min T(f)|)$ if it is represented in this manner.

Given incomparable sets MT, MF $(\subseteq \{0,1\}^n)$ such that $MT \cup MF \neq \emptyset$ and $T(MT) \cap F(MF) = \emptyset$, the partial function g is defined by

$$g(v) = \begin{cases} 1, & v \in T(MT) \\ 0, & v \in F(MF) \\ \text{unknown}, & \text{otherwise.} \end{cases}$$

If MT and MF of g satisfy $MT \subseteq \min T(f)$ and $MF \subseteq \max F(f)$ for some (complete) positive function f, then g is called a *partial function of* f. The set of unknown vectors of g is denoted by $U(g)$, i.e.,

$$U(g) = \{0,1\}^n \setminus (T(MT) \cup F(MF)).$$

The *k-neighborhood* of g is defined by

$$N_k(g) = \{v \mid \| v - a \| \le k \text{ for some } a \in MT \cup MF\},$$

where $\| w \|$ denotes $\sum_{i=1}^n |w_i|$. The *latency* of g, $\lambda(g)$, is defined to be the integer k satisfying

$$N_{k-1}(g) \cap U(g) = \phi \text{ and } N_k(g) \cap U(g) \neq \phi.$$

As a special case, if $U(g) = \phi$ i.e., $g = f$, then $\lambda(g)$ is defined to be 0. $\lambda(g)$ is equivalently given by

$$\lambda(g) = \min\{\| u - a \| \mid a \in MT \cup MF, u \in U(g)\}.$$

Now let C_X be a subclass of positive functions. $C_X(n)$ denotes the set of functions in C_X with n variables. For $C_X(n)$, the *maximum latency* is defined by

$$\Lambda_X(n) = \max\{\lambda(g) \mid g \text{ is a partial function of } f \in C_X(n)\}.$$

If g is a partial function of $f \in C_X(n)$, then by definition there is no unknown vector if $N_{\Lambda_X(n)}(g) \cap U(g) = \phi$. That is, in order to find an unknown vector, we only need to search $\Lambda_X(n)$-neighborhood of g. Therefore, if a positive function f of n variables is known to belong to class $C_X(n)$, Step 2 of Algorithm IDENTIFY can be executed as follows.

2. Test if $N_{\Lambda_X(n)}(g) \subseteq T(MT) \cup F(MF)$, where g is the partial function defined by MT and MF. If so, output MT and MF, and halt. Otherwise, find an unknown vector $u \in N_{\Lambda_X(n)}(g) \setminus (T(MT) \cup F(MF))$ and go to 3.

The test of $N_{\Lambda_X(n)}(g) \subseteq T(MT) \cup F(MF)$ can be accomplished by checking if $v \ge a$ for some $a \in MT$ or $v \le b$ for some $b \in MF$, for every $v \in N_{\Lambda_{X(n)}}(g)$. This computation takes at most

$$n(|MT| + |MF|)\,|N_{\Lambda_X(n)}(g)| = n(|MT| + |MF|)^2\, n^{\Lambda_X(n)}$$

time. Therefore, we have the next result.

Theorem 2.1 Let g, defined by MT and MF, be a partial function of $f \in C_X(n)$. If $\Lambda_X(n)$ is constant, then the above Step 2 can be executed in polynomial time in n and $|MT| + |MF|$. (Therefore, problem EQ can be solved in polynomial time, and there is an incrementally polynomial algorithm to identify $f \in C_X(n)$.) □

3 Restricted classes of positive functions with constant maximum latencies

In this section, we show that there are some nontrivial special classes of positive functions, which have constant maximum latency. These classes are important in practice and theory (e.g., [5, 10, 13]).

3.1 2-monotonic positive functions and Δ-partial positive threshold functions

If functions f and g satisfy $g(a) \leq f(a)$ for any $a \in \{0,1\}^n$, then we denote $g \leq f$. If $g \leq f$ and there exists a vector a satisfying $g(a) = 1$ and $f(a) = 0$, we denote $g < f$. An assignment A of binary values 0 or 1 to k variables $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ is called a *k-assignment*, and is denoted by

$$A = (x_{i_1} \leftarrow a_1, x_{i_2} \leftarrow a_2, \ldots, x_{i_k} \leftarrow a_k),$$

where each of $a_1, a_2, \ldots a_k$ is either 1 or 0. Let the complement of A, denoted by $\bar{A}$, represent the assignment obtained from A by complementing all the 1's and 0's in A. When a function f of n variables and a k-assignment A are given,

$$f_A = f_{(x_{i_1} \leftarrow a_1, x_{i_2} \leftarrow a_2, \ldots, x_{i_k} \leftarrow a_k)}$$

denotes the function of $(n-k)$ variables obtained by fixing variables $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ as specified by A.

Let f be a function of n variables. If either $f_A \leq f_{\bar{A}}$ or $f_A \geq f_{\bar{A}}$ holds for every k-assignment A, then f is said to be *k-comparable*. If f is k-comparable for every k such that $1 \leq k \leq m$, then f is said to be *m-monotonic*. (For more detailed discussion on these topics, see [10] for example.) In particular, f is 1-monotonic if $f_{(x_i \leftarrow 1)} \geq f_{(x_i \leftarrow 0)}$ or $f_{(x_i \leftarrow 1)} \leq f_{(x_i \leftarrow 0)}$ holds for any $i \in \{1, 2, \ldots, n\}$. A function f is positive if and only if f is 1 -monotonic and $f_{(x_i \leftarrow 1)} \geq f_{(x_i \leftarrow 0)}$ holds for all i.

Now consider a 2-assignment $A = (x_i \leftarrow 1, x_j \leftarrow 0)$. If

$$f_A \geq f_{\bar{A}} \text{ (resp. } f_A > f_{\bar{A}})$$

holds, this is denoted $x_i \succeq_f x_j$ (resp. $x_i \succ_f x_j$). Variables x_i and x_j are said to be *comparable* if either $x_i \succeq_f x_j$ or $x_i \preceq_f x_j$ holds. When $x_i \succeq_f x_j$ and $x_i \preceq_f x_j$ hold simultaneously, it is denoted as $x_i \approx_f x_j$. If f is 2-monotonic, this binary relation $\succeq_f$ over the set of variables is known to be a total preorder [10]. A 2-monotonic positive function f of n variables is called *regular* if

$$x_1 \succeq_f x_2 \succeq_f \cdots \succeq_f x_n.$$

Any 2-monotonic positive function becomes regular by permuting variables. Let

C_P: class of all positive functions,

C_{2M}: class of 2-monotonic positive functions.

Theorem 3.1 Class C_{2M} satisfies

$$\Lambda_{2M}(n) = 1.$$

Proof. Assume that a 2-monotonic positive function f is regular without loss of generality, and that g is a partial function of f defined by MT and MF. Assume that $N_1(g) \cap U(g) = \phi$ and $U(g) \neq \phi$. Take a $u \in \max U(g)$, where $\max U(g)$ is the set of maximal unknown vectors (i.e., $u + e_j \in T(MT)$ for all $j \in OFF(u)$). Let $j = \max\{i \mid i \in OFF(u)\}$. There exists $a \in MT$ such that $a \leq u + e_j$. Then $a - e_j \in F(MF)$ by assumption $N_1(g) \cap U(g) = \phi$. Therefore, there exists $b \in MF$ such that $b \geq a - e_j$. For any $l \in OFF(u) \setminus \{j\}$,

$$a - e_j + e_l \in T(f) \subseteq T(MT) \cup U(g)$$

by regularity of f, and hence $b \not\geq a - e_j + e_l$, i.e., $b \leq u$. (i) If $b = u$, then $u \in F(MF)$ which is a contradiction. (ii) If $b < u$, then $u \in T(MT)$ by $N_1(g) \cap U(g) = \phi$, which is also a contradiction. □

The 2-monotonicity was originally introduced in conjunction with threshold functions (e.g., [10]), where a positive function f is *threshold* if there exist $n+1$ nonnegative real numbers $w_1, w_2, \ldots, w_n$ and t such that:

$$f(x) = \begin{cases} 1, & \text{if } \sum w_i x_i \geq t \\ 0, & \text{if } \sum w_i x_i < t. \end{cases}$$

As $w_i \geq w_j$ implies $x_i \succeq_f x_j$ and $w_i = w_j$ implies $x_i \approx_f x_j$, a threshold function is always 2-monotonic, although the converse is not true [10]. Therefore, Theorem 3.1 tells class C_{TH} of positive threshold functions satisfies

$$\Lambda_{TH}(n) = 1.$$

Next, we generalize the concept of threshold functions by introducing some margin in the threshold value. A positive function f is called a *Δ-partial threshold function* [10] if f is represented by

$$f(x) = \begin{cases} 1, & \text{if } \sum w_i x_i \geq t + \alpha \\ 0, & \text{if } \sum w_i x_i < t - \alpha \\ 0 \text{ or } 1, & \text{otherwise,} \end{cases}$$

where $w_i\,(i = 1, 2, \ldots, n)$, t, Δ are nonnegative real numbers, and

$$\alpha = \Delta \min_i w_i.$$

In this definition, the value $f(x)$ in the case of "otherwise" can be arbitrary, provided that the resulting f is positive. Let

$C_{\Delta PTH}$: class of Δ-partial positive threshold functions.

For this class, we have the next result.

Theorem 3.2 [9] Class $C_{\Delta PTH}$ satisfies

$$\Lambda_{\Delta PTH}(n) \leq \lceil \Delta \rceil + 1. \quad \Box$$

3.2 Matroid functions

For a given vector $v \in \{0,1\}^n$, we use notations $ON(v) = \{j \mid v_j = 1\}$ and $OFF(v) = \{j \mid v_j = 0\}$. A positive function f is called a *matroid function* if for each $v, w \in \min T(f)$ and each $i \in ON(v) \setminus ON(w)$, there exists a $j \in ON(w) \setminus ON(v)$ such that $v - e_i + e_j \in \min T(f)$. In other words, $M = (E, \mathcal{F})$ forms a *matroid* [13], where $E = \{1, 2, \ldots, n\}$ and $\mathcal{F} = \{ON(v) \mid v \leq a \text{ for some } a \in \min T(f)\}$. Let

C_{MAT}: the class of matroid functions.

Theorem 3.3 [9] Class C_{MAT} of matroid functions satisfies

$$\Lambda_{MAT}(n) = \begin{cases} 1, & n = 1, 2, 3 \\ 2, & n \geq 4. \quad \Box \end{cases}$$

3.3 k-tight positive functions

A positive function f is called *k-tight* if it satisfies

$$\max\{\| a - b \| \mid a \in \min T(f), b \in \max F(f) \text{ and } a - e_j \leq b \text{ for some } j \in ON(a)\} \leq k,$$

where k is a positive integer. Let

C_{kTI}: class of k-tight positive functions.

For example, a positive threshold function with $w_{\max} \leq k w_{\min}$ is always k-tight, where $w_{\max} = \max_i w_i$ and $w_{\min} = \min_i w_i$, since for any $a \in \min T(f)$, $j \in ON(a)$ and $i_l \in OFF(a)$ $(l = 1, 2, \ldots, k)$,

$$\sum_{i=1}^n w_i a_i - w_j + \sum_{l=1}^k w_{i_l} \geq \sum w_i a_i - w_{max} + k w_{min} \geq \sum w_i a_i \geq t,$$

i.e., $a - e_j + \sum_{l=1}^k e_{i_l} \in T(f)$. To introduce other types of k-tight functions, define the *rank* of a set $S \subseteq \{0,1\}^n$ by $r(S) = \max\{\| x \| \mid x \in S\}$ and the *anti-rank* by $ar(S) = \min\{\| x \| \mid x \in S\}$, respectively. Then a positive function f satisfying one of the following conditions is k-tight.

(i) $|r(\max F(f)) - ar(\min T(f))| \leq k - 2$.
(ii) $ar(\min T(f)) \geq n - k + 1$.
(iii) $r(\max F(f)) \leq k - 1$.

These types of functions are discussed in [5] and other papers.

Theorem 3.4 [9] Class C_{kTI} of k-tight positive functions satisfies

$$\Lambda_{kTI}(n) \leq k. \quad \Box$$

4 General positive functions and positive k-DNF functions

Here we consider the class C_P of all positive functions, and

C_{kDNF}: class of positive k-DNF functions,

where a positive function f is a *positive k-DNF function* if $\| v \| \leq k$ for all $v \in \min T(f)$. It turns out that these classes do not have constant maximum latency.

$$
\begin{array}{ccc}
 & \begin{array}{ccccc} \overbrace{\hphantom{11\cdots1}}^{k-1} & \overbrace{\hphantom{00\cdots0}}^{k-1} & \overbrace{\hphantom{00\cdots0}}^{k-1} & \hphantom{\cdots} & \overbrace{\hphantom{00\cdots0}}^{k-1}\end{array} & \\
S = & \left(\begin{array}{ccccc}
11\cdots1 & 00\cdots0 & \cdots & \cdots & 00\cdots0 \\
11\cdots1 & 00\cdots0 & \cdots & \cdots & 00\cdots0 \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
11\cdots1 & 00\cdots0 & \cdots & \cdots & 00\cdots0 \\ \hline
00\cdots0 & 11\cdots1 & 00\cdots0 & \cdots & 00\cdots0 \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
00\cdots0 & 11\cdots1 & 00\cdots0 & \cdots & 00\cdots0 \\ \hline
\cdots & \cdots & \cdots & \cdots & \cdots \\ \hline
00\cdots0 & 00\cdots0 & 00\cdots0 & \cdots & 11\cdots1 \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
00\cdots0 & 00\cdots0 & 00\cdots0 & \cdots & 11\cdots1
\end{array}\right) &
\begin{array}{l}
\left.\begin{array}{c} \\ \\ \\ \\ \end{array}\right\}(\alpha-1)(k-1) \\
\left.\begin{array}{c} \\ \\ \\ \end{array}\right\}(\alpha-1)(k-1) \\
\vdots \\
\left.\begin{array}{c} \\ \\ \\ \end{array}\right\}(\alpha-1)(k-1)
\end{array}
\end{array}
$$

As noted before, this tells that the existence of an incrementally polynomial identification algorithm cannot be concluded from our approach. However, it dose not imply the nonexistence of such algorithm, and in fact it is known [5] that class C_{kDNF} has an incrementally polynomial identification algorithm, which is based on different idea.

Theorem 4.1 [8, 9] Class C_P of general positive functions satisfies

$$\lfloor n/4 \rfloor + 1 \leq \Lambda_P(n) \leq \lceil n/2 \rceil. \ \Box$$

For our purpose, the lower bound is more interesting, and it was shown in [9] by construction. We omit its description because the proof of Theorem 4.2 below also contains a similar construction. Also we conjecture $\Lambda_P(n) = \lfloor n/4 \rfloor + 1$, since $\Lambda_P(n) \leq \lfloor n/4 \rfloor + 1$ can be shown if we add a rather weak assumption on the set of unknown vectors $U(g)$ [8].

Theorem 4.2 Class C_{kDNF} satisfies, for $n \geq 4(k-1)$,

$$\Lambda_{kDNF}(n) \geq (k-1)\lfloor \sqrt{\frac{n}{(k-1)}} \rfloor - k + 2.$$

Proof. For $k = 1$, it is clear that $\Lambda_{1DNF}(n) \geq 1$. For $k \geq 2$, we provide an example of g with $\lambda(g) = (k-1)\lfloor \sqrt{\frac{n}{(k-1)}} \rfloor - k + 2$ for $n \geq 4(k-1)$. Let $\alpha = \lfloor \sqrt{\frac{n}{(k-1)}} \rfloor$, where α satisfies $\alpha \geq 2$. Then

$$\sqrt{\frac{n}{(k-1)}} \geq \alpha, \quad \text{i.e.,} \quad n \geq \alpha^2(k-1).$$

Therefore, let $n = \alpha^2(k-1) + \beta$, where β is a nonnegative integer. Now define a $\alpha(\alpha-1)(k-1) \times n$ matrix:

$$X = \left(\begin{array}{c|c|c} S & I_{\alpha(\alpha-1)(k-1)} & O_{\alpha(\alpha-1)(k-1)\times\beta} \end{array} \right),$$

where S is the $\alpha(\alpha-1)(k-1) \times \alpha(k-1)$ matrix as above. Here, I_j is the $j \times j$ unit matrix, and $O_{i\times j}$ is the $i \times j$ zero matrix. Define f by

$$\min T(f) = (\text{ the set of rows of matrix } X),$$

and a partial function g of f by $MT = \min T(f)$ and $MF = \max F(f) \setminus \{u\}$, where

$$u = (\underbrace{1,1,\cdots,1}_{\alpha(k-1)}, \underbrace{0,0,\cdots,0}_{\alpha(\alpha-1)(k-1)}, \underbrace{1,1,\cdots,1}_{\beta}).$$

Then $u + e_j \in T(MT)$ for any $j \in OFF(u)$ since $u \in \max F(f)$ and $MT = \min T(f)$. Moreover, $u - e_j \in F(MF)$ For any $j \in \{1, 2, \ldots, \alpha(k-1)\}$, and $u - \sum_{j\in S} e_j \notin F(MF)$, where $S = \{n-\beta+1, n-\beta+2, \ldots, n\}$. In other words,

$$U(g) = \{ \ (\underbrace{1,1,\cdots,1}_{\alpha(k-1)}, \underbrace{0,0,\cdots,0}_{\alpha(\alpha-1)(k-1)}, \underbrace{*,*,\cdots,*}_{\beta}) \ \},$$

where $*$ stands for 0 or 1. It is not difficult to see that $\| a - w \| = (\alpha-1)(k-1) + 1$ for every $a \in MT$, and $w \in U(g)$ and $\| b - w \| = (\alpha -$

1)(k − 1) + 1 for every $b \in MF$ and $w \in U(g)$. Therefore, its latency is

$$\begin{aligned}\lambda(g) &= (\alpha-1)(k-1)+1 \\ &= (k-1)\left\lfloor\sqrt{\frac{n}{(k-1)}}\right\rfloor - k + 2. \quad \square\end{aligned}$$

5 Discussion

In this paper, we introduced the maximum latency as a measure for the difficulty to find an unknown vector. Several interesting subclasses of positive functions have constant maximum latency. It would be important to find other subclasses of positive functions with constant maximum latency. Of course, the ultimate goal is to develop a polynomial time identification algorithm for general positive functions (or to disprove its existence) by some new tools.

Acknowledgments

The authors are grateful to the valuable comments given by Associate Professor H. Nagamochi and the members of our laboratory. This research was partially supported by the Scientific Grant-in-Aid from Ministry of Education, Science and Culture of Japan.

References

[1] D. Angluin, Queries and concept learning, *Machine Learning*, 2 (1988) 319-342.

[2] J. C. Bioch and T. Ibaraki, Complexity of identification and dualization of positive Boolean functions, *RUTCOR Research Report RRR 25-93*, Rutgers University, 1993.

[3] E. Boros, P. L. Hammer, T. Ibaraki and K. Kawakami, Identifying 2-monotonic positive Boolean functions in polynomial time, *ISA'91 Algorithms*, edited by W. L. Hsu and R. C. T. Lee, Springer Lecture Notes in Computer Science, 557 (1991) 104-115.

[4] Y. Crama, P. L. Hammer and T. Ibaraki, Cause-effect relationships and partially defined boolean functions, *Annals of Operations Research*, 16 (1988) 299-326.

[5] T. Eiter and G. Gottlob, Identifying the minimal transversals of a hypergraph and related problems, *Technical Report CD-TR 91/16*, Christial Doppler Labor für Expertensysteme, Technische Universität Wien, January 1991.

[6] T. Ibaraki and T. Kameda, A Theory of coteries: Mutual exclusion in distributed systems, *IEEE Trans. on Parallel and Distributed Systems*, 4 (1993) 779-794.

[7] D. S. Johnson, M. Yannakakis and C. H. Papadimitriou, On generating all maximal independent sets, *Information Processing Letters*, 27 (1988) 119-123.

[8] K. Makino and T. Ibaraki, The maximum latency of partially defined positive Boolean functions, *Trans. IEICE*, J76-D-1 (1993) 409-416 (in Japanese).

[9] K. Makino and T. Ibaraki, The maximum latency and identification of positive Boolean functions, Technical Report, Kyoto University, in preparation.

[10] S. Muroga, *Threshold Logic and Its Applications*, Wiley-Interscience, 1971.

[11] R. Reiter, A theory of Diagnosis from first principles, *Artificial Intelligence*, 32 (1987) 57-95.

[12] L. G. Valiant, A theory of the learnable, *Communications of the ACM*, 27 (1984) 1134-1142.

[13] D. J. A. Welsh, *Matroid Theory*, Academic Press, 1976.